Algorithmic Complexity Project

Casey Haynes

April 2018

1 Abstract

The following summary of my findings will be categorized in a bulleted list by function. All functions were tested with at least five different input sizes, and each input was tested three times. It is worth noting that the data points plotted in gnuplot were the average of the three data points found in the data tables for each function/input pair. They were not recorded in the data section, but are shown on each graph. The data section showcases the raw data collected, as well as predictions of complexity, and the graphs as an appendix.

2 Summary

1. Function 1:
   - My initial prediction: $O(n^2)$
   - The Actual Complexity: $\mathbf{O(n^3)}$
   - The function was found to have a complexity of O(n^3), supported by a fit of g(x) = cx^3, with c = 3.099 x $10^{-}9$ and an Asymptotic Standard Error = 2.844%
   - See graph in Data Section
2. Function 2:
   - My initial prediction: $O(n^2)$
   - The Actual Complexity: $\mathbf{O(2^n)}$
   - The function was found to have a complexity of O(2^n), supported by a fit of g(x) = $c2^x$, with c = 4.507 x $10^{-}9$ and an Asymptotic Standard Error = 1.851%
   - See graph in Data Section
3. Function 3 Sorted (Best or Worst Case):
   - My initial prediction: $O(n)$
   - The Actual Complexity: **O(n)**
   - The function was found to have a complexity of O(n), supported by a fit of g(x) = cx, with c = 6.068 x $10^{-}8$ and an Asymptotic Standard Error = 1.129%
   - See graph in Data Section
4. Function 3 Random (Avg. Case):
   - My initial prediction: $O(n)$
   - The Actual Complexity: **O(n)**
   - The function was found to have a complexity of O(n), supported by a fit of g(x) = cx, with c = 2.109 x $10^{-}7$ and an Asymptotic Standard Error = 0.720%
   - See graph in Data Section
5. Function 3 Reverse (Worst or Best Case):
   - My initial prediction: $O(n)$
   - The Actual Complexity: **O(n)**

- The function was found to have a complexity of O(n), supported by a fit of g(x) = cx, with c = 6.155 x $10^{-}8$ and an Asymptotic Standard Error = 0.724%
- See graph in Data Section

6. Function 4:
   - My initial prediction: $O(n)$
   - The Actual Complexity: **O(n)**
   - The function was found to have a complexity of O(n), supported by a fit of g(x) = cx, with c = 8.675 x $10^{-}8$ and an Asymptotic Standard Error = 0.678%
   - See graph in Data Section
7. Function 5:
   - My initial prediction: $O(logn)$
   - The Actual Complexity: **O(logn)**
   - The function was found to have a complexity of O(logn), supported by a fit of g(x) = $clog(x)$, with c = 0.004 and an Asymptotic Standard Error = 0.843%
   - See graph in Data Section
8. Function 6 Sorted (Best or Worst Case):
   - My initial prediction: $O(nlogn)$
   - The Actual Complexity: **O(n^2)**
   - The function was found to have a complexity of O(n^2), supported by a fit of g(x) = cx^2, with c = 0.003 and an Asymptotic Standard Error = 0.176%
   - See graph in Data Section
9. Function 6 Random (Avg. Case):
   - My initial prediction: $O(nlogn)$
   - The Actual Complexity: **O(n^2)**
   - The function was found to have a complexity of O(n^2), supported by a fit of g(x) = cx^2, with c = 0.004 and an Asymptotic Standard Error = 0.830%
   - See graph in Data Section
10. Function 6 Reverse (Worst or Best Case):
    - My initial prediction: $O(nlogn)$
    - The Actual Complexity: **O(n)**
    - The function was found to have a complexity of O(n), supported by a fit of g(x) = cx, with c = 0.003 and an Asymptotic Standard Error = 2.477%
    - There's a slight anomaly here: the runtime for the reverse sorted array is significantly shorter than those for sorted and random. The fact that the number of CPU ticks increases uniformly with each increase in input size makes me believe that the algorithm is designed to handle a reverse sorted array. i.e. it's sorting algorithm is taking more time in the other two instances. Here, we're seeing it run without having to sort, leading to a different level of complexity.
    - See graph in Data Section
11. Function 7:
    - My initial prediction: $O(n!)$
    - The Actual Complexity: **O(n!)**
    - The function was found to have a complexity of O(n!), supported by a fit of g(x) = $cgamma(x+1)$, with c = 1.201 and an Asymptotic Standard Error = 0.047%
    - See graph in Data Section

3 Data

Function 1

Guess: $O(n^2)$

Trial	n-value	time (seconds)	time (CPU ticks)
1	10	0.000	4
2		0.000	3
3		0.000	3
1	100	0.003	3108
2		0.003	3055
3		0.003	3084
1	200	0.024	24757
2		0.025	25204
3		0.024	24852
1	300	0.083	83467
2		0.083	83538
3		0.083	83790
1	400	0.197	197144
2		0.198	198607
3		0.195	195541
1	500	0.382	382008
2		0.382	382359
3		0.385	385846
1	600	0.686	686694
2		0.679	679253
3		0.659	659989
1	700	1.050	1050434
2		1.050	1050246
3		1.061	1061614
1	800	1.567	1567431
2		1.578	1578919
3		1.599	1599390
1	900	2.239	
2		2.259	
3		2.263	
1	1000	3.044	
2		3.120	
3		3.092	

	Trial	n-value	time (seconds)	time (CPU ticks)	
	1	20	0.005		
	2		0.004		
	3		0.005		
	1	21	0.008		
	2		0.009		
	3		0.009		
	1	22	0.018		
	2		0.017		
	3		0.017		
	1	23	0.037		
	2		0.037		
	3		0.038		
	1	24	0.074		
	2		0.075		
Function 2	3		0.074		Guess: $O(n^2)$
	1	25	0.152		
	2		0.149		
	3		0.149		
	1	26	0.298		
	2		0.298		
	3		0.296		
	1	27	0.597		
	2		0.598		
	3		0.599		
	1	28	1.197		
	2		1.202		
	3		1.191		
	1	29	2.435		
	2		2.417		
	3		2.432		

Function 3

Trial	n-value	time (s) SORT	time (s) RAND	time (s) REV
1	1 x 10^6	0.053	0.185	0.052
2		0.053	0.186	0.051
3		0.054	0.185	0.052
1	2 x 10^6	0.112	0.386	0.114
2		0.112	0.392	0.112
3		0.111	0.386	0.110
1	3 x 10^6	0.117	0.606	0.181
2		0.177	0.595	0.178
3		0.177	0.606	0.184
1	4 x 10^6	0.233	0.807	0.231
2		0.238	0.820	0.238
3		0.231	0.833	0.241
1	5 x 10^6	0.296	1.037	0.304
2		0.301	1.054	0.306
3		0.295	1.056	0.310
1	6 x 10^6	0.373	1.248	0.378
2		0.373	1.262	0.384
3		0.367	1.260	0.375
1	7 x 10^6	0.436	1.499	0.430
2		0.432	1.449	0.431
3		0.434	1.468	0.437
1	8 x 10^6	0.484	1.699	0.491
2		0.489	1.670	0.490
3		0.487	1.703	0.493
1	9 x 10^6	0.550	1.938	0.549
2		0.553	1.967	0.560
3		0.555	1.918	0.563

Guess: $\underline{O(n)}$

	Trial	n-value	time (seconds)	time (CPU ticks)	
	1	1 x 10^7	0.882		
	2		0.877		
	3		0.820		
	1	1.1 x 10^7	0.896		
	2		0.948		
	3		0.956		
	1	1.2 x 10^7	0.997		
	2		1.000		
	3		1.013		
	1	1.3 x 10^7	1.081		
	2		1.097		
	3		1.108		
	1	1.4 x 10^7	1.211		
	2		1.189		
Function 4	3		1.185		Guess: O(n)
	1	1.5 x 10^7	1.293		
	2		1.293		
	3		1.292		
	1	1.6 x 10^7	1.359		
	2		1.396		
	3		1.396		
	1	1.7 x 10^7	1.504		
	2		1.528		
	3		1.519		
	1	1.8 x 10^7	1.568		
	2		1.575		
	3		1.612		
	1	1.9 x 10^7	1.650		
	2		1.683		
	3		1.699		

Function 5

Trial	n-value	time (seconds)	time (CPU ticks)
1	10^3	0.028	
2		0.027	
3		0.027	
1	10^4	0.035	
2		0.035	
3		0.034	
1	10^5	0.046	
2		0.047	
3		0.046	
1	10^6	0.054	
2		0.054	
3		0.055	
1	10^7	0.061	
2		0.062	
3		0.062	
1	10^8	0.070	
2		0.069	
3		0.069	
1	10^9	0.080	
2		0.080	
3		0.081	

Guess: $O(logn)$

	Trial	n-value	time (ticks) SORT	time (ticks) RAND	time (ticks) REV	
	1	5000	73272	77590	18	
	2		72284	78585	17	
	3		72707	78051	19	
	1	10000	295680	354861	37	
	2		287815	358994	36	
	3		285684	356089	37	
	1	15000	650135	855542	48	
	2		655329	846956	48	
Function 6	3		662813	853880	48	Guess: $O(nlogn)$
	1	20000	1170011	1556719	74	
	2		1161409	1581724	64	
	3		1166468	1570137	63	
	1	25000	1822908	2476267	79	
	2		1842396	2565370	118	
	3		1817216	2475243	80	
	1	30000	2618711	3622532	94	
	2		2683894	3655092	95	
	3		2643130	3676022	107	

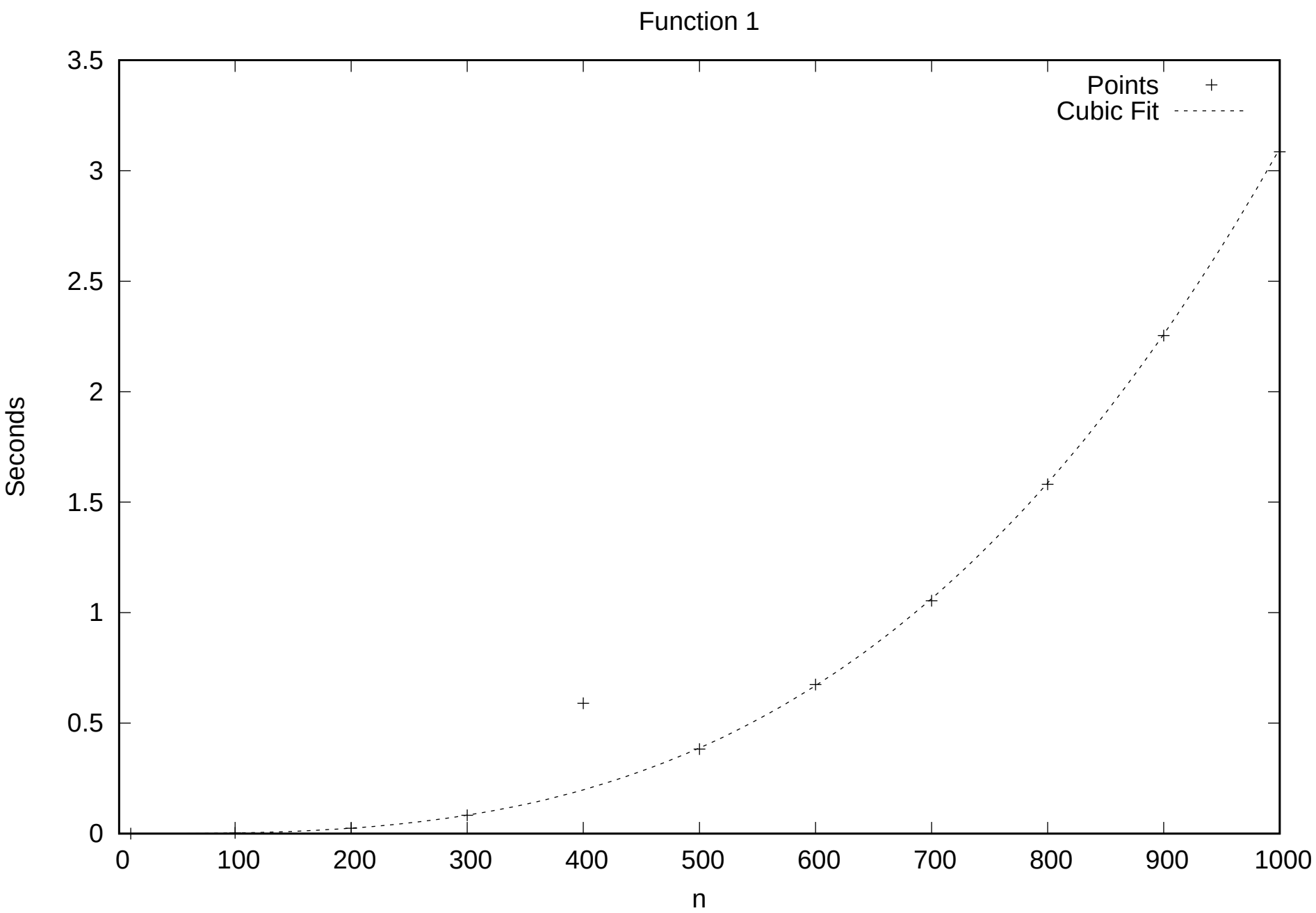
Function 7

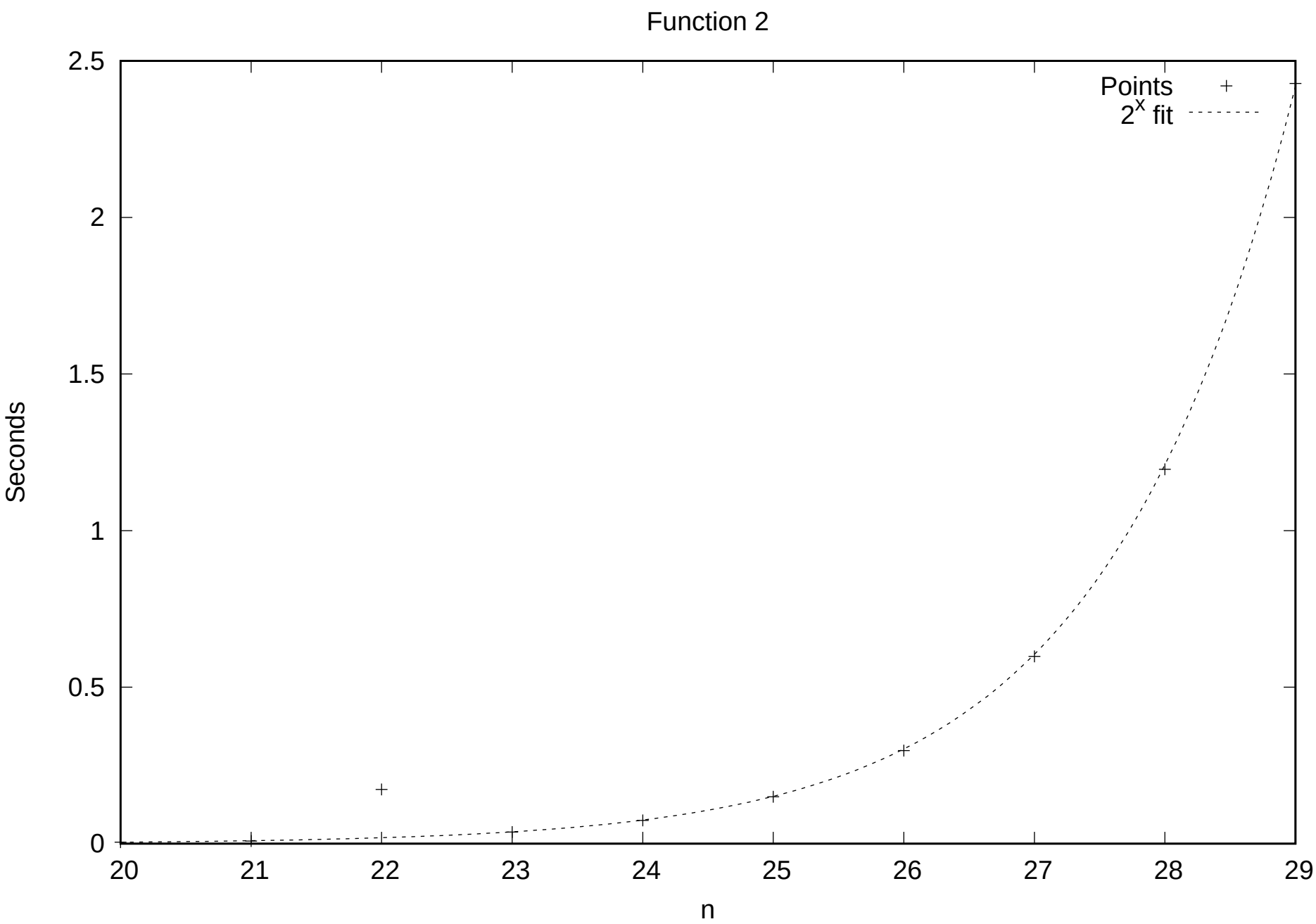
Trial	n-value	time (seconds)	time (CPU ticks)
1	8	0.000	415
2		0.000	468
3		0.000	444
1	9	0.003	
2		0.003	
3		0.005	
1	10	0.043	
2		0.043	
3		0.042	
1	11	0.472	
2		0.480	
3		0.470	
1	12	5.789	
2		5.726	
3		5.740	

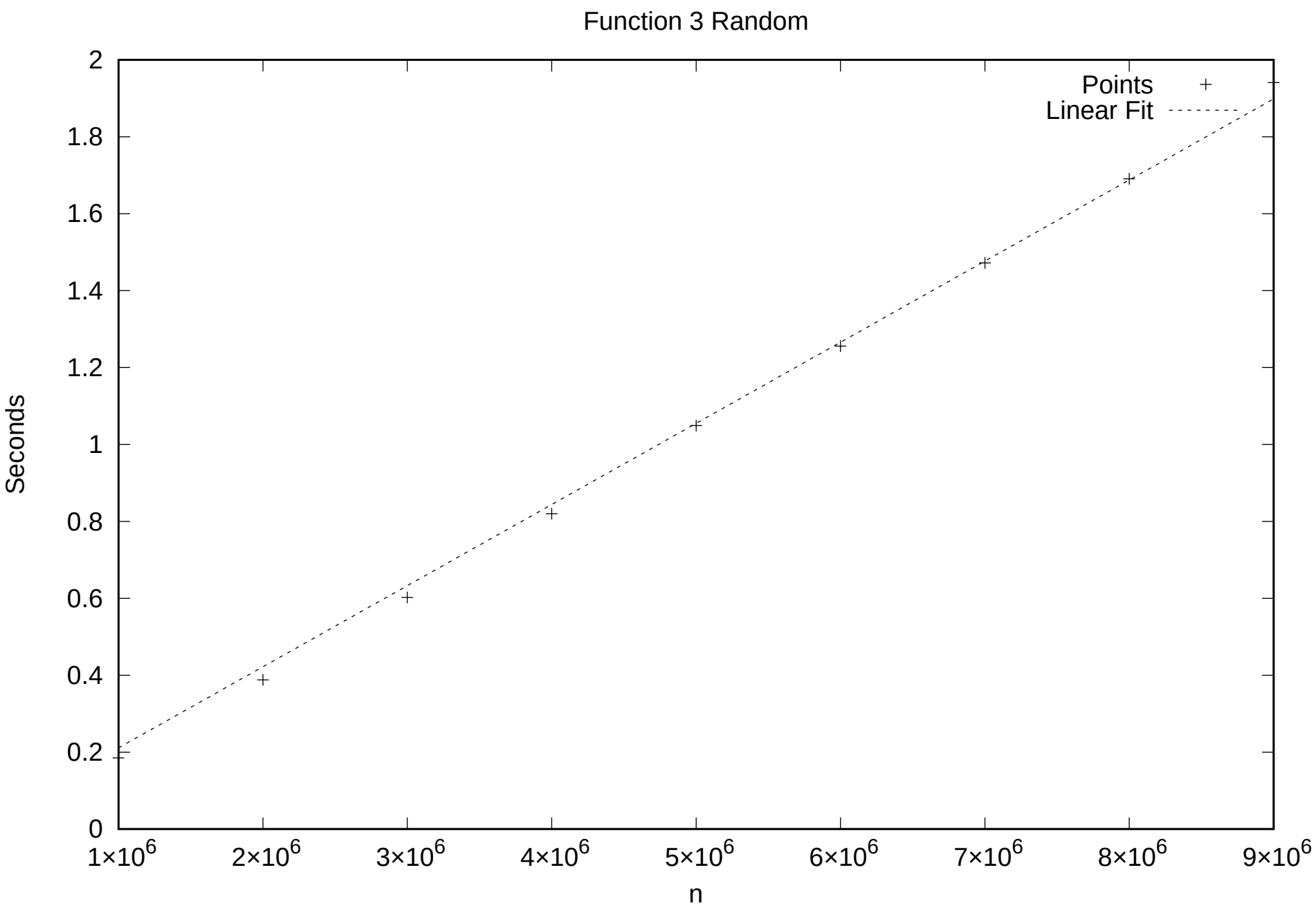
Guess: O(n!)

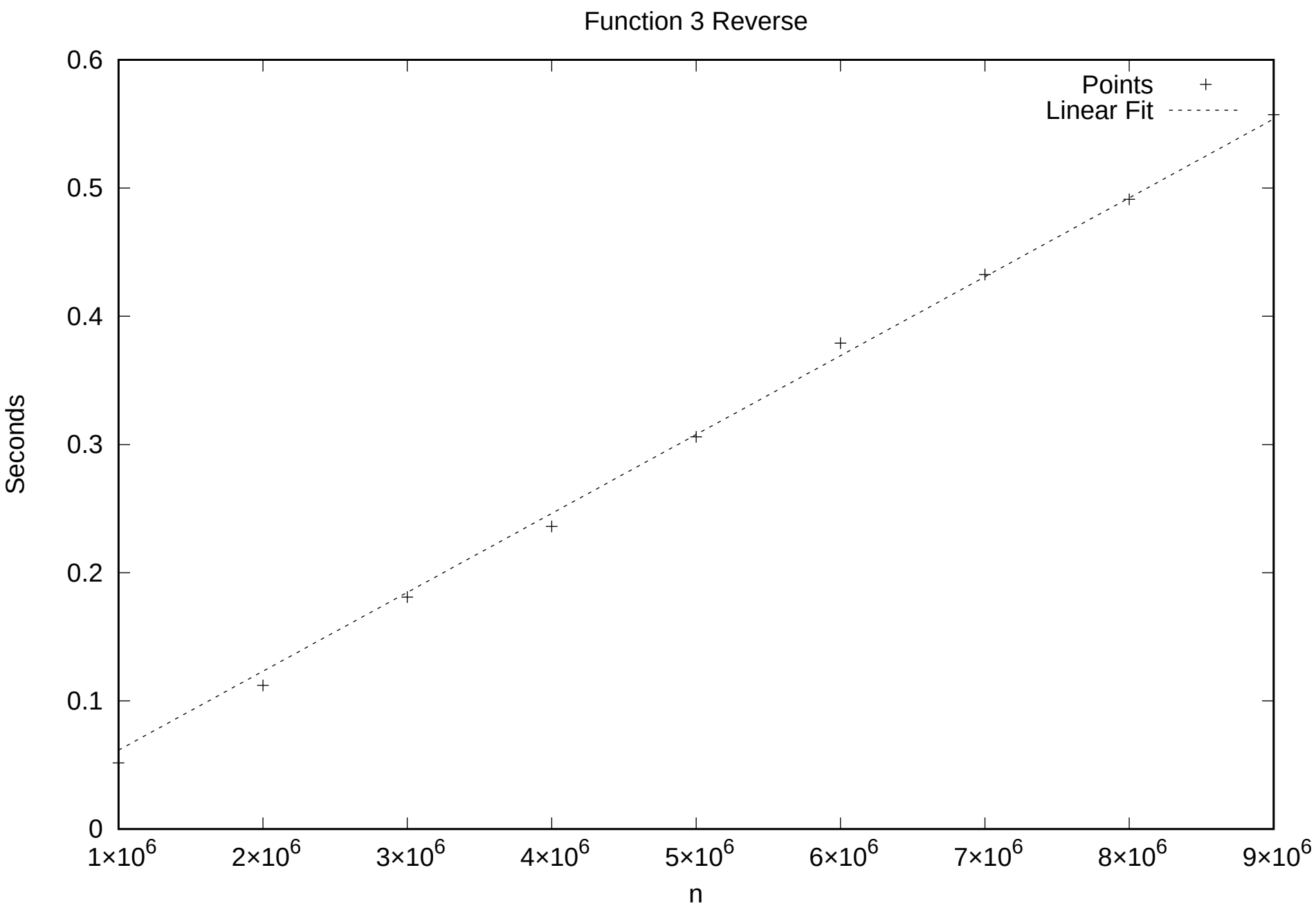
C-Values, Asymptotic Standard Errors, and Functions Fitted

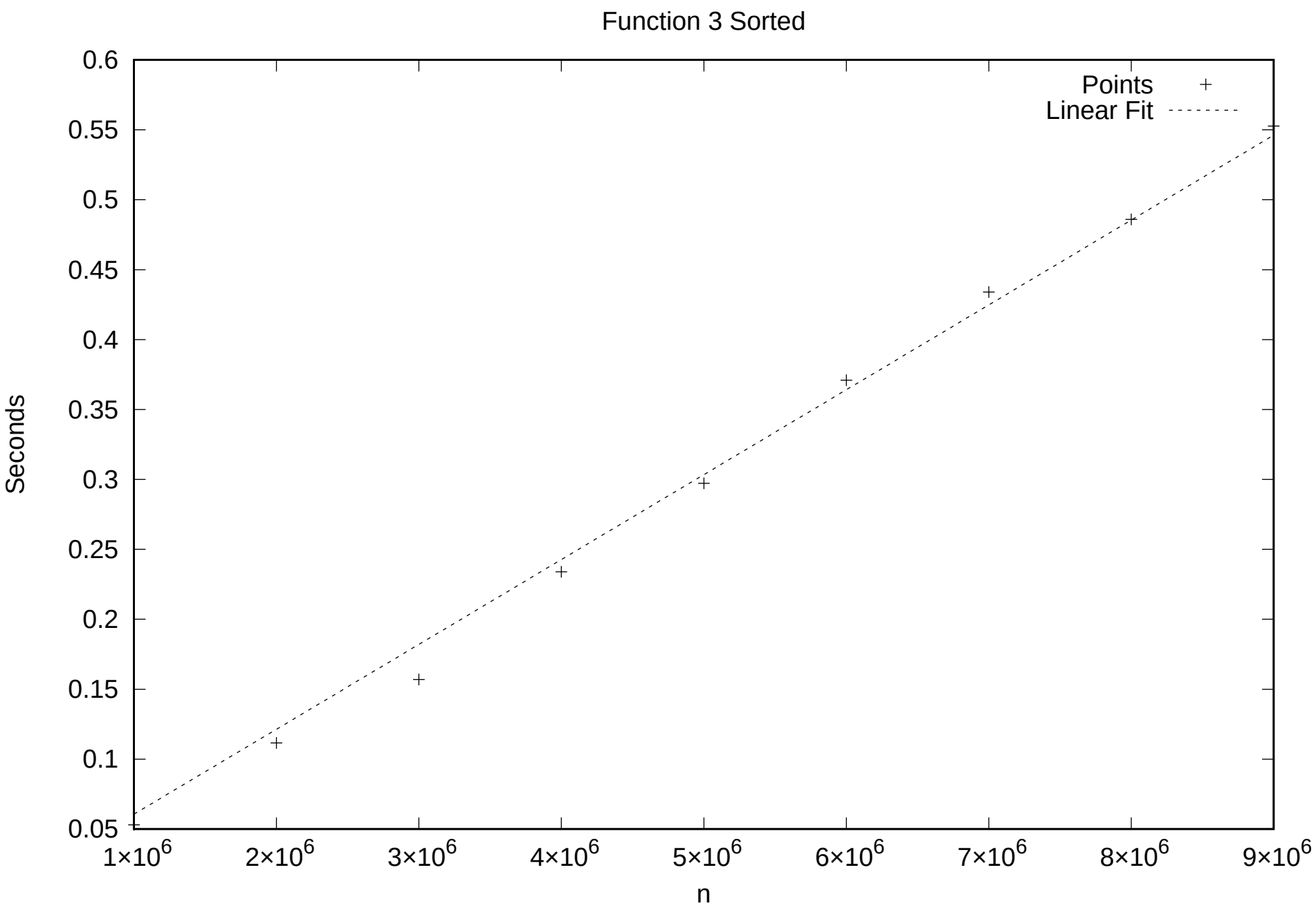
- Function 1:
  - c = 3.09996 x $10^{-}9$
  - Asymptotic Standard Error = 2.844 percent
  - x^3
- Function 2:
  - c = 4.50714 x $10^{-}9$
  - Asymptotic Standard Error = 1.851 percent
  - 2^x
- Function 3 sorted:
  - c = 6.06846 x $10^{-}8$
  - Asymptotic Standard Error = 1.129 percent
  - x
- Function 3 random:
  - c = 2.10972 x $10^{-}7$
  - Asymptotic Standard Error = 0.7204 percent
  - x
- Function 3 reverse:
  - c = 6.1547 x $10^{-}8$
  - Asymptotic Standard Error = 0.7236 percent
  - x
- Function 4:
  - c = 8.67484 x $10^{-}8$
  - Asymptotic Standard Error = 0.678 percent
  - x
- Function 5:
  - c = 0.00385576
  - Asymptotic Standard Error = 0.8433 percent
  - log(x)
- Function 6 Sorted:
  - c = 0.00293325
  - Asymptotic Standard Error = 0.1765 percent
  - x^2
- Function 6 Random:
  - c = 0.00401531
  - Asymptotic Standard Error = 0.8303 percent
  - x^2
- Function 6 Reverse:
  - c = 0.00342044
  - Asymptotic Standard Error = 2.477 percent
  - x
- Function 7:
  - c = 1.20065 x $10^{-}8$
  - Asymptotic Standard Error = 0.04717 percent
  - gamma(x+1) = x!

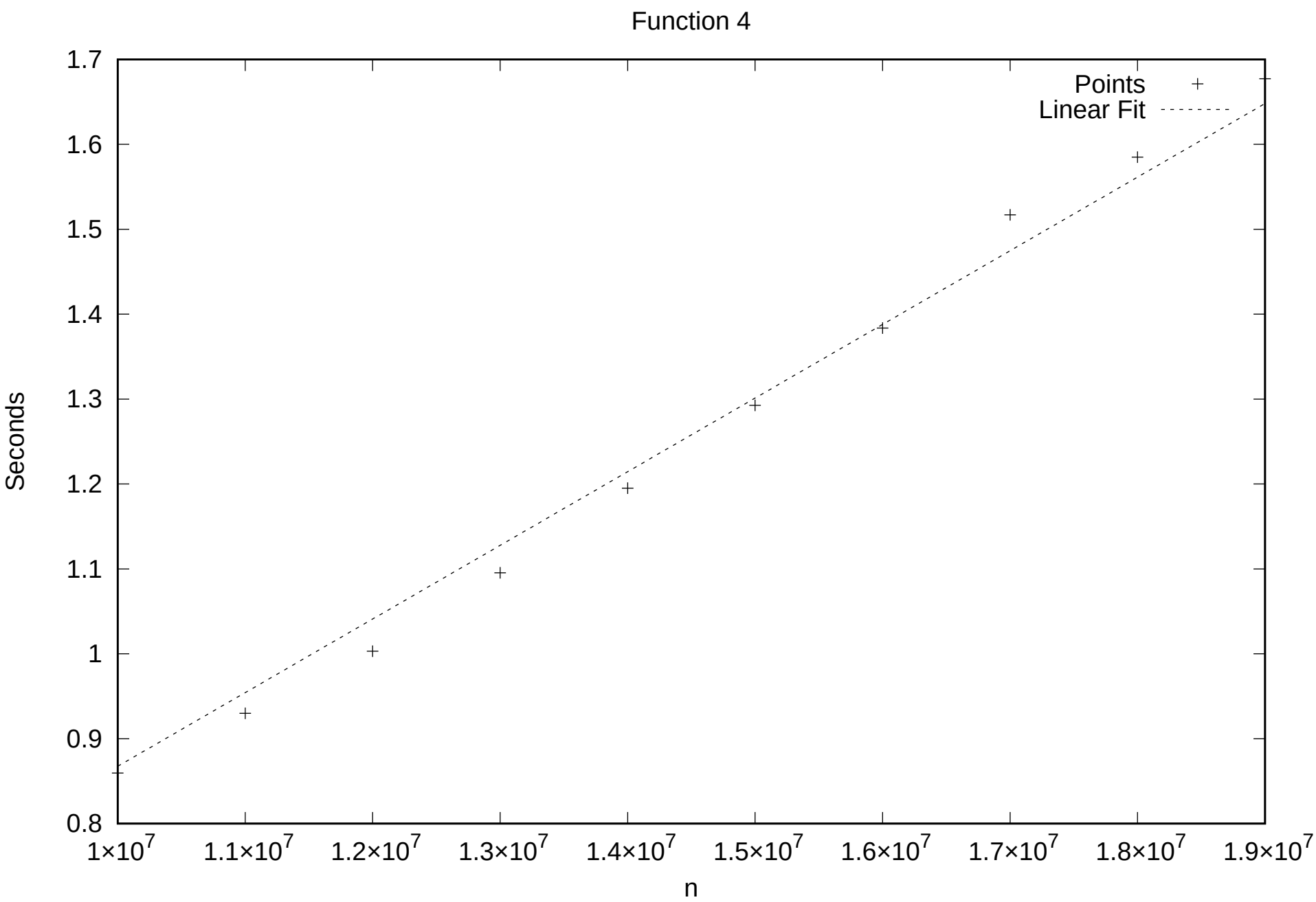

Function 1
Points
Cubic Fit
Seconds
n
0
0.5
1
1.5
2
2.5
3
3.5
0
100
200
300
400
500
600
700
800
900
1000

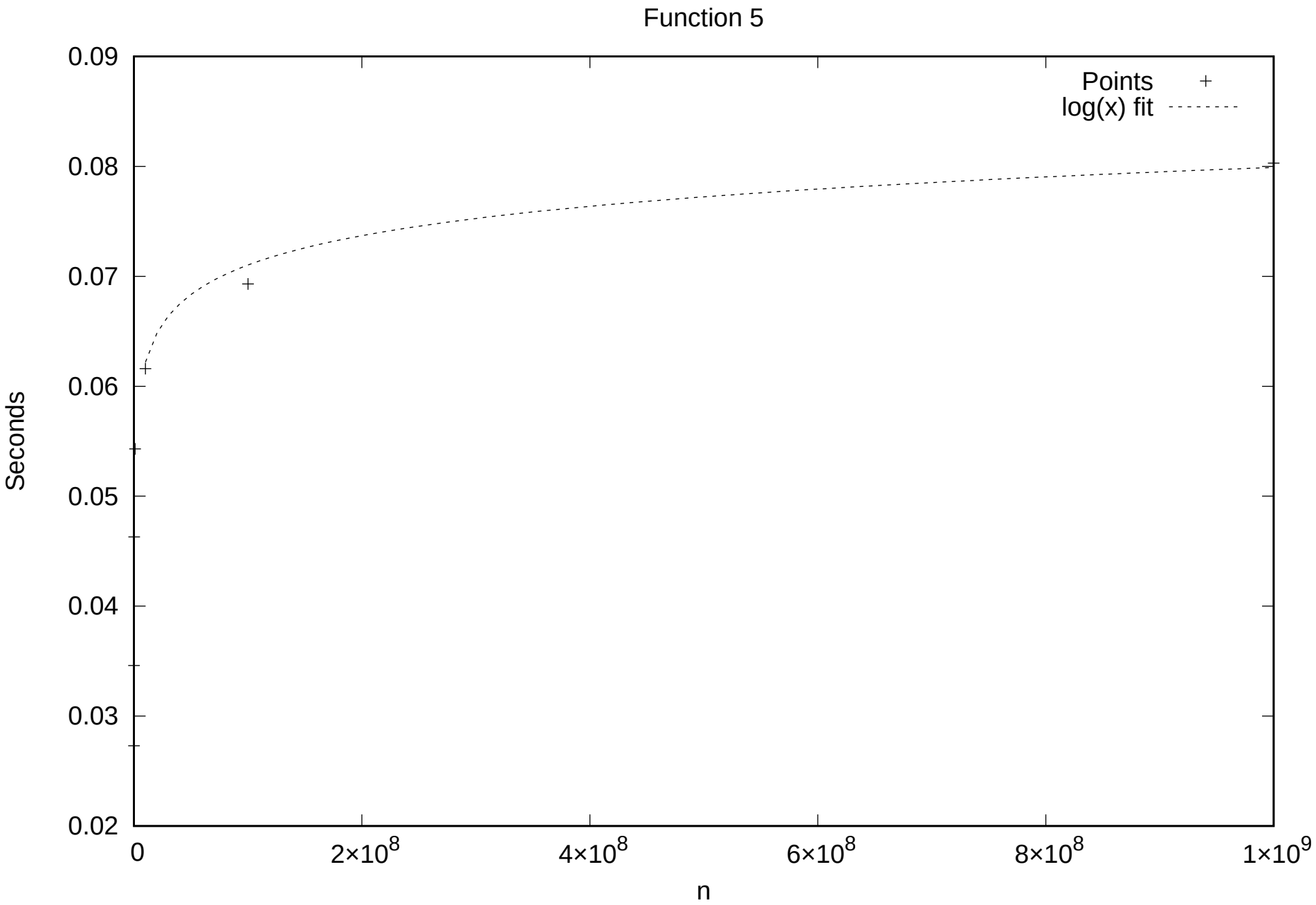

Function 2
Points
2^x fit
Seconds
n
2.5
2
1.5
1
0.5
0
20
21
22
23
24
25
26
27
28
29

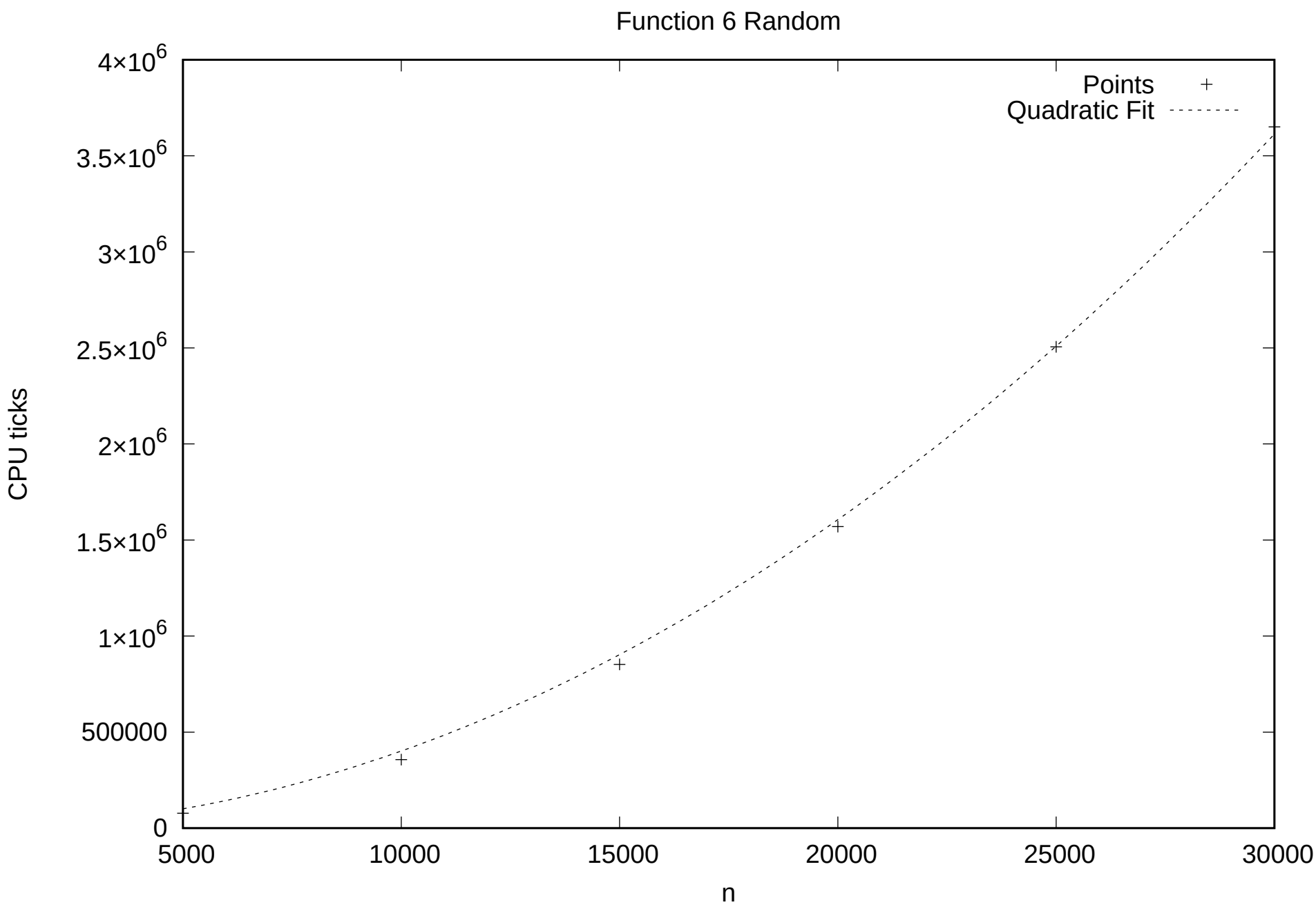

Function 3 Random
Points
Linear Fit
Seconds
n
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
1×10^6
2×10^6
3×10^6
4×10^6
5×10^6
6×10^6
7×10^6
8×10^6
9×10^6

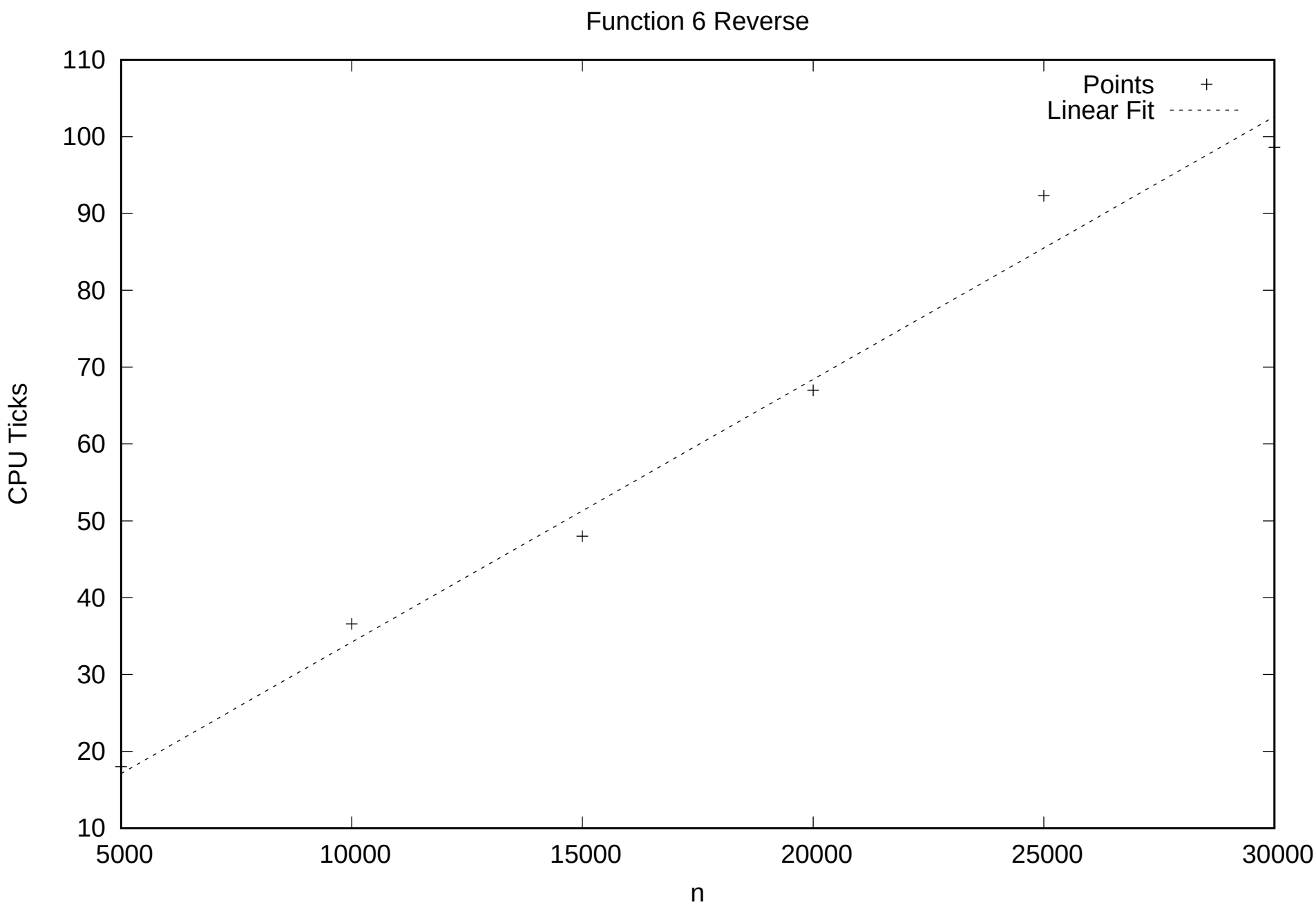

Function 3 Reverse
Points
Linear Fit
Seconds
n
0
0.1
0.2
0.3
0.4
0.5
0.6
1×10^6
2×10^6
3×10^6
4×10^6
5×10^6
6×10^6
7×10^6
8×10^6
9×10^6

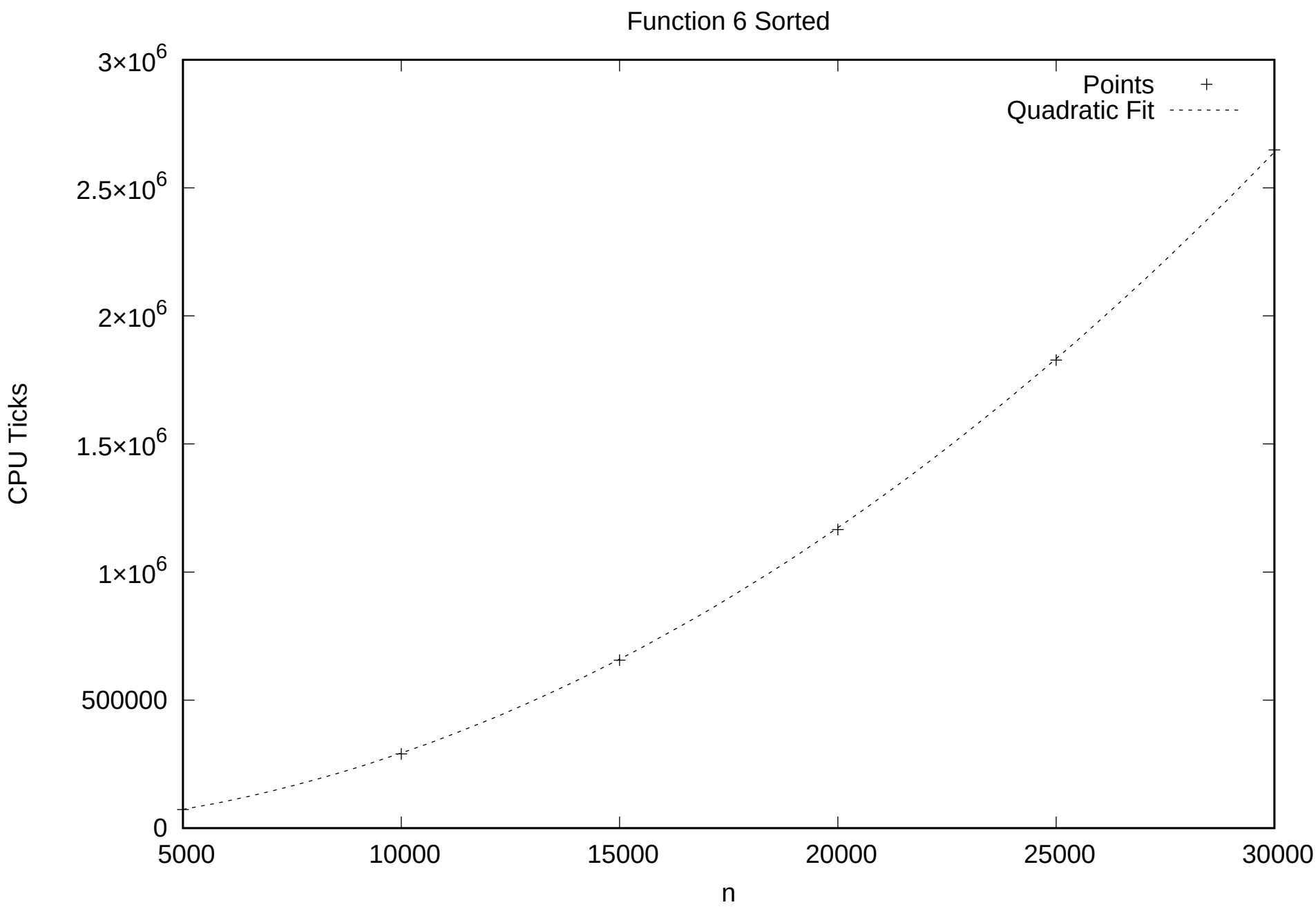

Function 3 Sorted
Points
Linear Fit
Seconds
n
0.6
0.55
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
1×10^6
2×10^6
3×10^6
4×10^6
5×10^6
6×10^6
7×10^6
8×10^6
9×10^6

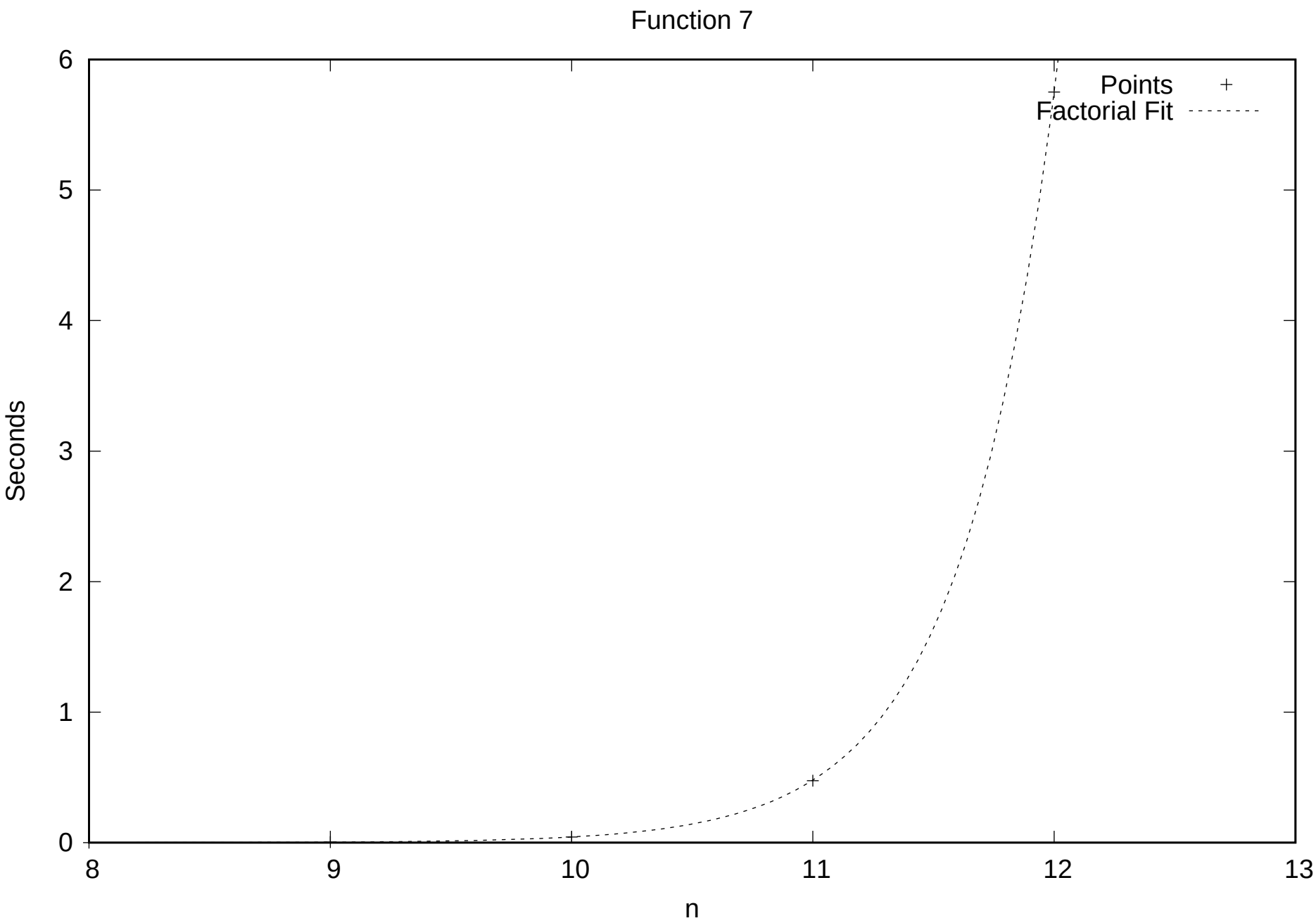

Function 4
Points
Linear Fit
Seconds
n
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1
0.9
0.8
1×10^7
1.1×10^7
1.2×10^7
1.3×10^7
1.4×10^7
1.5×10^7
1.6×10^7
1.7×10^7
1.8×10^7
1.9×10^7


Function 5
Points
log(x) fit
Seconds
n
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0
2×10^8
4×10^8
6×10^8
8×10^8
1×10^9


Function 6 Random
Points
Quadratic Fit
CPU ticks
n
4×10^6
3.5×10^6
3×10^6
2.5×10^6
2×10^6
1.5×10^6
1×10^6
500000
0
5000
10000
15000
20000
25000
30000


Function 6 Reverse
CPU Ticks
n
Points
Linear Fit
110
100
90
80
70
60
50
40
30
20
10
5000
10000
15000
20000
25000
30000


Function 6 Sorted
Points
Quadratic Fit
CPU Ticks
n
3×10^6
2.5×10^6
2×10^6
1.5×10^6
1×10^6
500000
0
5000
10000
15000
20000
25000
30000


Function 7
Points
Factorial Fit
Seconds
n
0
1
2
3
4
5
6
8
9
10
11
12
13